Project-Bragboard

Name: Lakshmi Prasanna

milestone-2:

Week 4 Overview — Shout-outs Feed & Department Filters

Objective:
During Week 4, the goal was to develop the **Shout-outs Feed**, a key component of the Brag Board, where employees can view and celebrate achievements across departments. The focus was on displaying all shout-outs, adding smart filters, and enabling optional media uploads.

Key Features Implemented:

1. **Display All Shout-outs on the Feed:**
   - Designed a dynamic **feed interface** showcasing every shout-out in chronological order.
   - Each post displays the **sender's name, department, message, and timestamp**.
   - Optimized for responsiveness and clarity across all screen sizes.
2. **Filter Functionality:**
   - Introduced **department-based filters** for better organization and navigation.
   - Users can filter shout-outs by:
     - **Information Technology (IT)**
     - **Finance**
     - **Human Resources (HR)**
   - Additionally, filters for **sender name** and **date range** were included for precise searching.
   - All filters are connected to FastAPI endpoints, ensuring smooth, real-time data fetching without page reloads.
3. **Attachments / Image Uploads (Optional):**
   - Integrated optional **attachment support** for image or document uploads.
   - Implemented **image preview** and file validation for format and size.
   - Managed uploads securely using FastAPI's UploadFile and backend storage handling.

Tech Stack Used:

- **Frontend:** React.js
- **Backend:** FastAPI (Python)
- **Database:** Postgresql
- **Storage:** Local/Cloud for media files

Outcome:
By the end of Week 4, the Brag Board featured a fully functional **interactive feed** with advanced filtering and media capabilities. This allowed employees from **IT, Finance, and HR**

departments to connect, appreciate, and recognize one another's achievements in an engaging way.

Common Errors Faced

- Faced CORS errors between frontend and backend connections.
- Fixed by enabling CORS middleware in FastAPI.
- Encountered file upload issues with invalid formats.
- Resolved by adding file type and size validation.

Project Setup and Environment Configuration

- **A monorepo structure was created, separating the project into a dedicated backend directory for the FastAPI application and a frontend directory for the React.js application.**
- **The backend environment was configured with a Python virtual environment and all necessary dependencies, including SQLAlchemy for database interactions.**
- **The frontend environment was initialized using Vite, and Tailwind CSS was installed and configured for efficient styling.**
- **A scalable folder structure was implemented for both backend and frontend to ensure maintainability:**
  - **Backend: models, schemas, crud, api**
  - **Frontend: pages, components, services**

Instructions for Running the Project

To run the BragBoard project, both the backend (FastAPI) and frontend (React) need to be set up and running. Follow these steps:

Backend (FastAPI)

1. **Open a terminal and navigate to the backend directory: cd backend**
2. **Create and activate a Python virtual environment:**
   - **Windows: python -m venv venv → venv\Scripts\activate**
   - **macOS/Linux: python -m venv venv → source venv/bin/activate**
3. **Install the required dependencies using pip install -r requirements.txt**
4. **Start the FastAPI server using: uvicorn app.main:app --reload**
   - **The backend will run at http://127.0.0.1:8000/docs**

Frontend (React with Vite)

1. **Open a terminal and navigate to the frontend directory: cd frontend**
2. **Install the required packages using npm install**
3. **Start the development server with npm run dev**
   - **The frontend will run at http://localhost:5173**

Project Repository

The source code for BragBoard is hosted on GitHub and can be accessed at the following link:

BragBoard Repository.